

Unsupervised Document Scaling with Quanteda

Kenneth Benoit and Paul Nulty

April 29, 2014

Loading Documents into Quanteda

One of the most common tasks

The quanteda package provides several functions for loading texts from disk into a quanteda corpus. In this example, we will load a corpus from a set of documents in a directory, where each document's attributes are specified in its filename. In this case, the filename contains the variables of interest, separated by underscores, for example:

2010_BUDGET_03_Joan_Burton_LAB.txt

Quanteda provides a function to create a corpus from a directory of documents like this. The user needs to provide the path to the directory, the names of the attribute types, and the character which separates the attribute values in the filenames:

```
library(quanteda)
dirname <- "~/Dropbox/QUANTESS/corpora/iebudgets/budget_2010/"
attNames <- c("year", "debate", "number", "firstname", "surname", "party")
ieBudgets <- corpusFromFilenames(dirname, c("year", "debate", "no", "fname", "speaker",
      "party"), sep = "_")
```

This creates a new quanteda corpus object where each text has been associated values for its attribute types extracted from the filename:

```
summary(ieBudgets)

## Corpus object contains 14 texts.
##
##               Texts  Types  Tokens  Sentences  year  debate
## 2010_BUDGET_01_Brian_Lenihan_FF.txt    1649    7720      390 2010  BUDGET
## 2010_BUDGET_02_Richard_Bruton_FG.txt     951    4035      222 2010  BUDGET
## 2010_BUDGET_03_Joan_Burton_LAB.txt    1473    5711      329 2010  BUDGET
## 2010_BUDGET_04_Arthur_Morgan_SF.txt    1455    6432      349 2010  BUDGET
## 2010_BUDGET_05_Brian_Cowen_FF.txt    1470    5835      262 2010  BUDGET
## 2010_BUDGET_06_Enda_Kenny_FG.txt    1059    3853      161 2010  BUDGET
## 2010_BUDGET_07_Kieran_ODonnell_FG.txt   609    2049      141 2010  BUDGET
## 2010_BUDGET_08_Eamon_Gilmore_LAB.txt   1088    3767      208 2010  BUDGET
## 2010_BUDGET_09_Michael_Higgins_LAB.txt  439    1132       49 2010  BUDGET
## 2010_BUDGET_10_Ruairi_Quinn_LAB.txt   413    1177       60 2010  BUDGET
## 2010_BUDGET_11_John_Gormley_Green.txt  362     919       49 2010  BUDGET
## 2010_BUDGET_12_Eamon_Ryan_Green.txt   482    1513       90 2010  BUDGET
## 2010_BUDGET_13_Ciaran_Cuffe_Green.txt  422    1140       48 2010  BUDGET
## 2010_BUDGET_14_Caoimhghin_OCaolain_SF.txt 1040    3614      194 2010  BUDGET
## no      fname  speaker party
## 14 Caoimhghin OCaolain  SF
```

```
## 13      Ciaran      Cuffe Green
## 12      Eamon      Ryan Green
## 11      John      Gormley Green
## 10      Ruairi      Quinn LAB
## 09      Michael    Higgins LAB
## 08      Eamon      Gilmore LAB
## 07      Kieran    ODonnell FG
## 06      Enda      Kenny FG
## 05      Brian      Cowen FF
## 04      Arthur      Morgan SF
## 03      Joan      Burton LAB
## 02      Richard    Bruton FG
## 01      Brian      Lenihan FF
##
## Source: /home/paul/Dropbox/code/quanteda/tutorials/scaling/* on x86_64 by paul.
## Created: Tue Apr 29 14:01:06 2014.
## Notes: NA.
```

In order to perform statistical analysis such as document scaling, we must extract a matrix containing the frequency of each word type from in document. In quanteda, we use the `dfm` function to produce such a matrix.¹

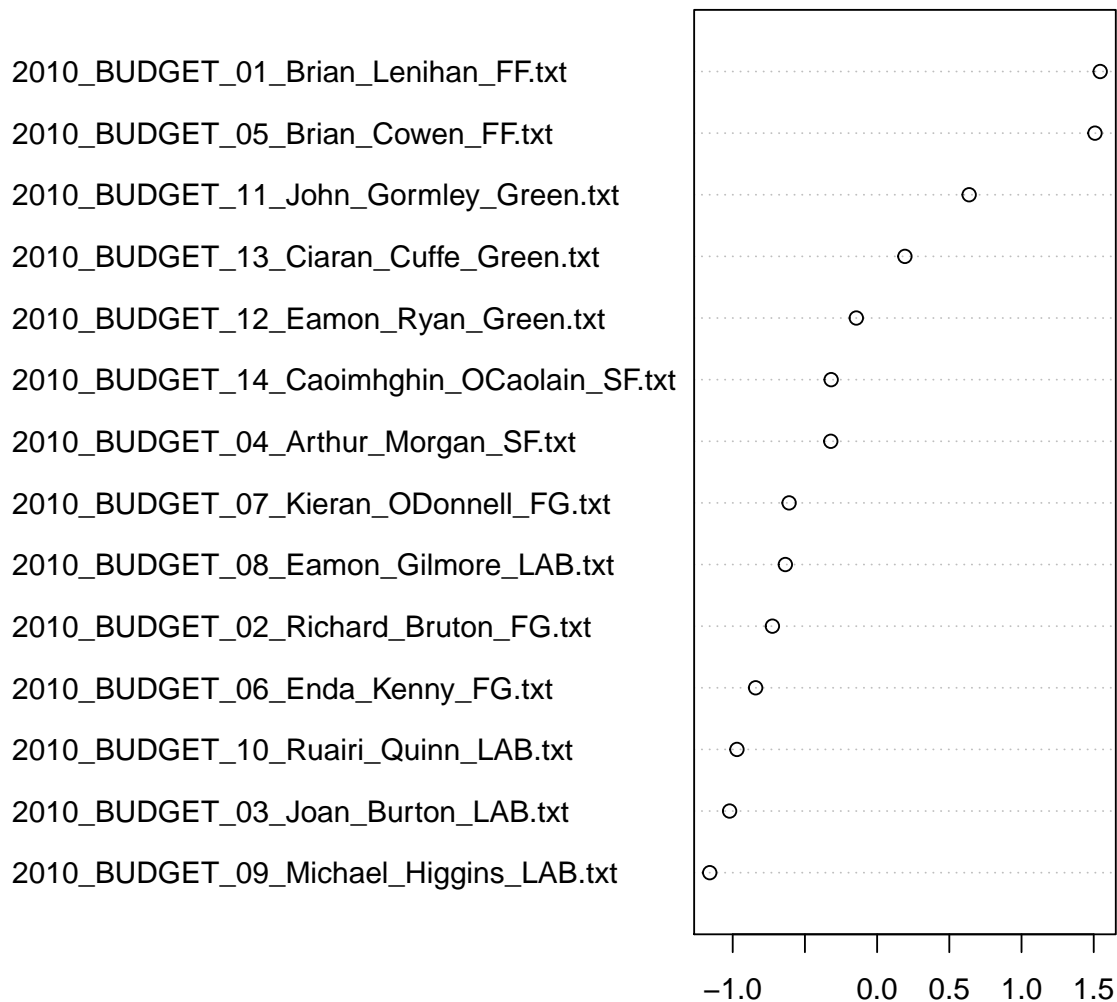
```
docMat <- dfm(ieBudgets)

## Creating dfm: ... done.
```

We can now score and plot the documents using a statistical scaling technique, for example correspondence analysis [Nenadic and Greenacre, 2007].

```
library(ca)
model <- ca(t(docMat), nd = 1)
dotchart(model$colcoord[order(model$colcoord[, 1]), 1], labels = model$colnames[order(model$colcoord[, 1])])
```

¹`dfm` stands for document-feature matrix — we say ‘feature’ instead of word, as it is sometimes useful to represent documents by features other than their word frequency.

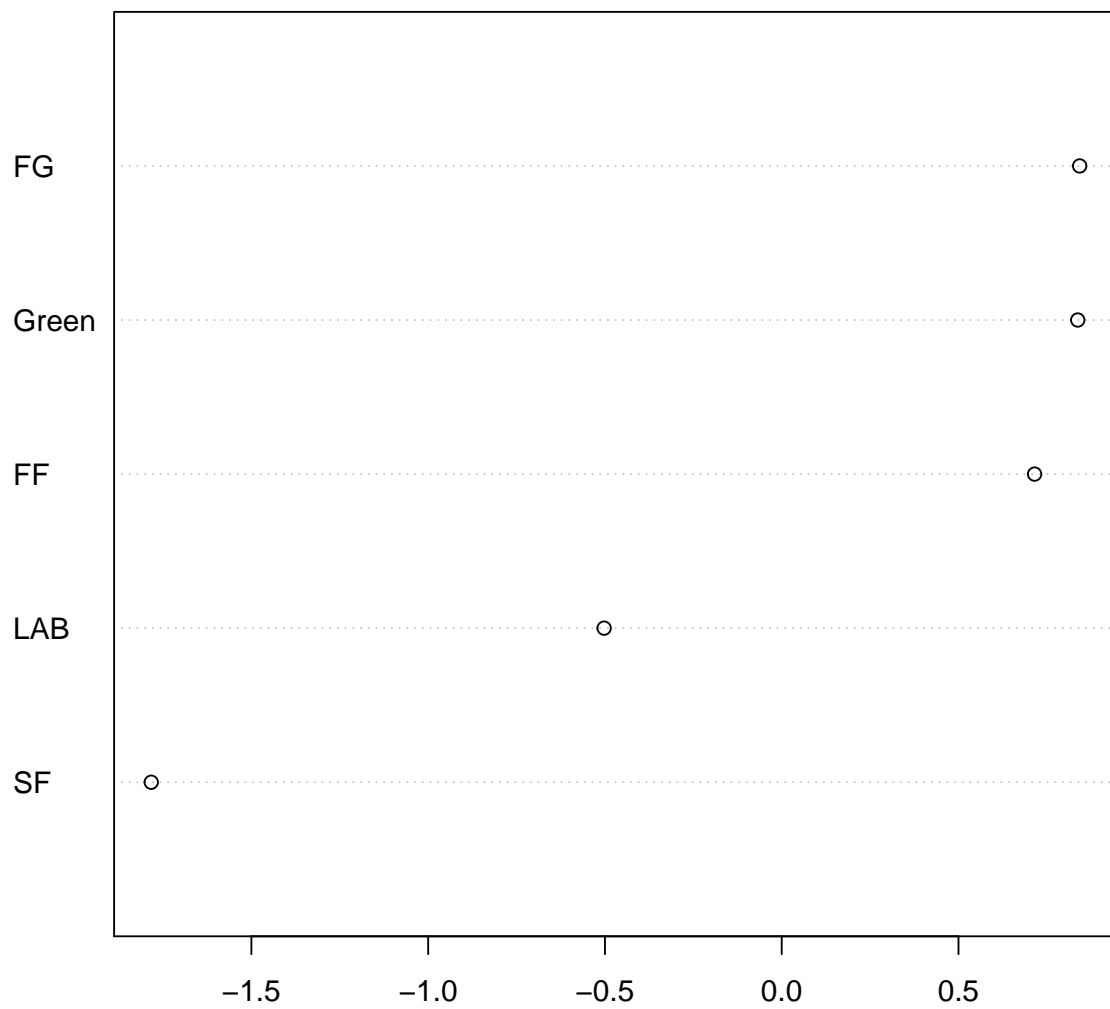


This plot indicates the position of each of the documents. We can group documents by their attribute values when creating the word-frequency matrix, which allows us to scale according to a particular party or year, for example

```
partyMat <- dfm(ieBudgets, group = "party")

## Creating dfm: ... aggregating by group: party...complete ... done.

partyModel <- ca(t(partyMat), nd = 1)
dotchart(partyModel$colcoord[order(partyModel$colcoord[, 1]), 1], labels = partyModel$colnames[order(partyModel$colcoord[, 1])])
```



References

Oleg Nenadic and Michael Greenacre. Correspondence analysis in *r*, with two-and three-dimensional graphics: The *ca* package. 2007.